Priors on the Variance in Sparse Bayesian Learning; the demi-Bayesian Lasso

Suhrid Balakrishnan
AT&T Labs Research
180 Park Avenue
Florham Park, NJ 07932
suhrid@research.att.com

David Madigan
Department of Statistics
Columbia University
New York, NY 10027
madigan@stat.columbia.edu

August 21, 2009

Abstract

We explore the use of proper priors for variance parameters of certain sparse Bayesian regression models. This leads to a connection between sparse Bayesian learning (SBL) models (Tipping, 2001) and the recently proposed Bayesian Lasso (Park and Casella, 2008). We outline simple modifications of existing algorithms to solve this new variant which essentially uses type-II maximum likelihood to fit the Bayesian Lasso model. We also propose an Elastic-net (Zou and Hastie, 2005) heuristic to help with modeling correlated inputs. Experimental results show the proposals to compare favorably to both the Lasso and traditional and more recent sparse Bayesian algorithms.

1 Introduction and Motivation

Sparse Bayesian Learning (SBL) using automatic relevance determination as typified by the Relevance Vector machine (Tipping, 2001), has proven to be a very effective and accurate method for a wide variety of regression and classification problems. The SBL paradigm performs parameter learning via

type-II maximum likelihood where a marginal data likelihood maximization provides the parameter estimates. Two related tracks, the Lasso (Tibshirani, 1996) and the Bayesian Lasso (Park and Casella, 2008), approach the estimation task in rather different ways. The Lasso considers regression and classification in the loss plus ℓ_1-regularization framework. The resulting optimization problem can also be viewed in the Bayesian setting as a maximum-*a-posteriori* (MAP) solution to a regression problem with parameters having individual Laplace (or double exponential) priors. The Bayesian Lasso instead makes use of the equivalence of a hierarchical Gaussian-Exponential prior to the Laplace prior, and conducts fully Bayesian inference (via Markov chain Monte Carlo or MCMC sampling algorithms) for parameter inference.

A number of recent papers have explored connections between these three approaches and our work is in that vein. For example Wipf and Nagarajan (2008) clearly delineates the connection between SBLs type-II maximum likelihood and MAP estimation, by showing that SBL's type-II maximum likelihood is equivalent to MAP estimation where the prior on the parameters is "non-factorial" (in other words, the prior depends on the input basis functions, and cannot be decomposed into independent terms involving each parameter). A natural question that arises is whether type-II maximum likelihood is an effective way to train the Bayesian Lasso model as well. This would have two advantages over the Bayesian Lasso. First, parameter estimates would be sparse, and second, the parameter estimates would be obtained by optimization and not by computationally more demanding MCMC.

2 Background and Notation

We consider SBL, the Lasso and the Bayesian Lasso in the context of the classical Gaussian linear regression modeling. Specifically, given a regressor matrix/feature dictionary $\mathbf{\Phi}$, an observation/response vector $\mathbf{y}$ and i.i.d. Gaussian noise/errors $\boldsymbol{\epsilon}$, we consider linear models of the form

$$\mathbf{y} = \mathbf{\Phi}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{1}$$

These assumptions lead to a likelihood of the form:

$$p(\mathbf{y}|\boldsymbol{\beta}, \sigma^2, \mathbf{\Phi}) = 2\pi\sigma^{2^{-N/2}} \exp\left\{-\frac{\|\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta}\|}{2\sigma^2}\right\}$$

where the dataset, $\mathcal{D}$ comprises N responses $\mathbf{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$ and the N x p design matrix $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]^{\mathrm{T}}$. The Gaussian noise distribution is mean zero and variance σ^2, $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}|0, \sigma^2 I)$ and the parameter vector $\boldsymbol{\beta}$ is p dimensional. We assume that the intercept parameter, if any, is estimated outside the estimation schemes discussed here (for example, by centering the response). Loosely speaking, the Lasso is the least "Bayesian" of three approaches while the Bayesian Lasso is the most Bayesian. SBL along with the "demi-Bayesian" approach we describe below are somewhere in between.

2.1 The Lasso

The Lasso formulation estimates $\boldsymbol{\beta}$ by solving the following convex optimization problem:

$$\min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})^{\mathrm{T}} (\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta}) + \rho \|\boldsymbol{\beta}\|_1$$

(ρ is a non-negative scalar regularization parameter). The Lasso optimization problem has a MAP-Bayesian interpretation as follows (Tibshirani, 1996). Assign each component β_j of $\boldsymbol{\beta}$ an independent Laplacian or double-exponential prior distribution with mean 0:

$$p(\beta_j|\rho_j) = \frac{\rho_j}{2} e^{-\rho_j|\beta_j|}, \rho_j > 0, j = 1, \ldots, p$$

with $p(\boldsymbol{\beta}) = \prod_j p(\beta_j)$ and all $\rho_j = \rho$. A prior of this form places high probability mass near zero and along individual component axes thereby promoting sparsity (see Figure 1). It also has heavier tails than a Gaussian distribution leading to some theoretical difficulties with regard to variable selection[1].

Now, in this setting, the Lasso optimization problem results in $\boldsymbol{\beta}$ estimates that correspond to the posterior mode estimates ($argmax_{\boldsymbol{\beta}} p(\boldsymbol{\beta}|\mathcal{D}, \rho)$). Predictions are then made using this point posterior mode. By contrast, fully Bayesian inference would typically integrate over the entire posterior distribution rather than conditioning on a specific value. In fact, while the posterior mode is an optimal point estimate under zero-one loss, there is no particular reason to expect such a loss function to be reasonable in any particular application. Nonetheless, the Lasso has provided excellent predictive performance in many applications (Genkin et al., 2007).

[1]It is now well-known that the Lasso does not possess an "Oracle Property," typically failing to set enough components of $\boldsymbol{\beta}$ to zero. See, for example, Zou (2006).

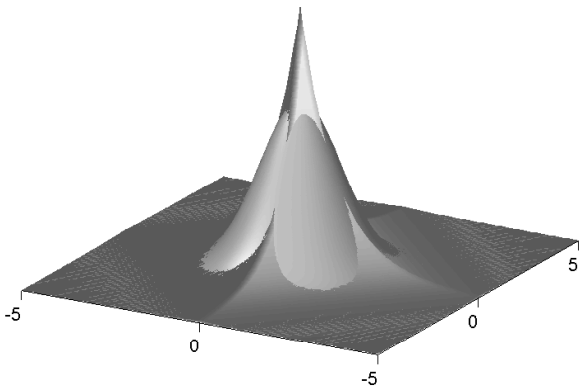

Figure 1: A superposition of a standard (zero mean, unit variance) two dimensional Gaussian distribution, and a Laplace distribution ($\rho = 1$). The figure highlights the higher probability mass the Laplace assigns along the axes and at zero as well as its heavier tails.

2.2 Sparse Bayesian Learning

An alternative sparse linear modeling approach was proposed by Tipping (2001) in his work on the relevance vector machine (referred to as SBL here). In this line of work, a zero-mean Gaussian prior is assumed for each of the regression parameters:

$$p(\boldsymbol{\beta}|\boldsymbol{\gamma}) = \prod_{j=1}^{p} \mathcal{N}(\beta_j|0, \gamma_j), \gamma_j > 0, j = 1, \ldots, p, \tag{2}$$

where crucially, each unknown weight has a separate non-negative hyper-parameter γ_j controlling it's variance (with $\boldsymbol{\gamma}$ being the p vector of these hyperparameters). In the learning procedure, sparsity is achieved if certain γ_j are set to zero. A further hierarchical specification of the hyperparameters (for both the variance of the weights and the noise) completes the prior specification, with $p(\gamma) = \prod_j Gamma(\gamma_j|a, b)$ and $p(\sigma^2) = Gamma(\sigma^2|c, d)$. In the RVM and further works however, these priors are specified as flat and hence improper priors (a, b, c, d=0), an important point of difference with what we propose.

Learning in the SBL paradigm involves exact posterior inference for the predictions, where the hyperparameters are chosen to maximize the marginal data likelihood. The literature refers to this procedure as type-II maximum

likelihood or evidence maximization (Mackay, 1992; Berger, 1980). Equivalently, SBL minimizes:

$$-\log \int p(\mathbf{y}|\boldsymbol{\beta})p(\boldsymbol{\beta}|\gamma)d\boldsymbol{\beta} = \log|\Sigma_y| + \mathbf{y}^{\mathrm{T}}\Sigma_y^{-1}\mathbf{y} \tag{3}$$

where $\Sigma_y = \sigma^2 I + \boldsymbol{\Phi}\Gamma\boldsymbol{\Phi}^{\mathrm{T}}$ and $\Gamma = \mathrm{diag}[\boldsymbol{\gamma}]$ (see Tipping 2001 or Wipf and Nagarajan 2008). This optimization leads to some $\boldsymbol{\gamma}_*$, which then leads to the posterior distribution of the weights $p(\boldsymbol{\beta}|\mathcal{D}, \boldsymbol{\gamma}_*, \sigma^2) = \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}, \Sigma)$. Here, $\boldsymbol{\mu} = \Gamma_*\boldsymbol{\Phi}^{\mathrm{T}}\Sigma_{y*}^{-1}\mathbf{y}$ and $\Sigma = \Gamma - \Gamma\boldsymbol{\Phi}^{\mathrm{T}}\Sigma_{y*}^{-1}\boldsymbol{\Phi}\Gamma$ [2]. The expression for the posterior mean $\boldsymbol{\mu}$, further emphasizes how if $\gamma_{*,j} = 0$, the corresponding β_j is also zero and removed from the model. Finally, the predictive density for a new/test point $\phi(\mathbf{x}_t)$ integrates over the posterior density of $\boldsymbol{\beta}$ leading to a closed-form Gaussian expression:

$$\begin{aligned} p(y_t|\mathcal{D}, \boldsymbol{\gamma}_*, \sigma^2, \phi(\mathbf{x}_t)) &= \mathcal{N}(y_t|m_{y_t}, \sigma_t). \\ m_{y_t} &= \boldsymbol{\mu}^{\mathrm{T}}\phi(\mathbf{x}_t), \\ \sigma_t^2 &= \sigma^2 + \phi(\mathbf{x}_t)^{\mathrm{T}}\Sigma\phi(\mathbf{x}_t). \end{aligned} \tag{4}$$

We note that SBL can be shown to be equivalent to Gaussian process regression under particular restrictions - see for example Tipping (2001).

The objective function in the SBL optimization problem (in Equation 3) is multi-modal, non-convex, and has fixed points at sparse solutions. Various algorithms have been proposed in the literature for obtaining local minima (Tipping, 2001; Wipf and Nagarajan, 2008; Tipping and Faul, 2003; Mackay, 1992).

2.3 The Bayesian Lasso

The Bayesian Lasso (Park and Casella, 2008) starts with the data model of Equation 1 and the same Gaussian prior for the weights as in SBL (Equation 2). The hierarchical prior model differs slightly from that of SBL insofar as the variance parameters are assumed to be drawn from an exponential distribution with rate hyperparameter p-vector $\boldsymbol{\lambda}$, instead of a gamma distribution,i.e.:

$$p(\boldsymbol{\gamma}|\boldsymbol{\lambda}) = \prod_{j=1}^{p} \frac{\lambda_j}{2} \exp -\frac{\lambda_j\gamma_j}{2}, \lambda_j > 0, j = 1, \ldots, p.$$

[2] The expressions are modeled on the Wipf and Nagarajan (2008) paper, and are equivalent to the ones in the RVM paper where the notation is slightly different.

The reason why this relates to the Lasso and sparse learning, is because this particular form of hierarchical prior results in a Laplace prior on $\boldsymbol{\beta}$ after marginalizing out $\boldsymbol{\gamma}$ ($p(\boldsymbol{\beta}) = \int p(\boldsymbol{\beta}|\boldsymbol{\gamma})p(\boldsymbol{\gamma}|\boldsymbol{\lambda})d\boldsymbol{\gamma}$). This result derives from the representation of the Laplace distribution as a scaled mixture of Gaussians with an exponential mixing density(Park and Casella, 2008):

$$\frac{\sqrt{a}}{2}e^{-\sqrt{a}|z|} = \int_0^\infty \frac{1}{\sqrt{2\pi s}}e^{-z^2/(2s)}\frac{a}{2}e^{-as/2}ds, a > 0.$$

Inference in the Bayesian Lasso is carried out in a fully Bayesian manner via posterior simulation. Exploiting closed form marginal distribution calculations, Park and Casella (2008) outline a Gibbs sampler that can be used to draw samples from the posterior distribution $p(\boldsymbol{\beta}|\mathcal{D})$ (they also propose various techniques to estimate/set/sample from the hyperparameter distribution). While this represents a satisfying Bayesian solution, MCMC sampling poses a significant obstacle in terms of the size of the applications this technique can reasonably be expected to handle. In addition, the Bayesian Lasso does not yield a sparse solution unless ad-hoc rules are used to threshold components of $\boldsymbol{\beta}$ that are small *a posteriori*. Other minor sampling related drawbacks include difficulty in assessing convergence of the MCMC sampler, and tuning of the sampling algorithm itself.

3 The demi-Bayesian Lasso

With the above background in place we turn to our proposals. To circumvent the computational complexities associated with the MCMC sampling required for the Bayesian Lasso, we propose fitting the Bayesian Lasso model through a type-II maximum likelihood procedure (i.e., by maximizing the marginal data likelihood). Conceptually, this inherits the benefits of the SBL framework and alleviates the corresponding sampling associated problems. We now find hyperparameters via optimization and not sampling (thus greatly expanding the dimensionality of models that can be learnt efficiently), the resultant posterior distribution is analytically tractable (Gaussian), and sparse models for prediction are obtained without thresholding the posterior distribution. Of course, the flip side is that first, this proposal, like SBL, is less than fully Bayesian, and second, also like SBL, it results in a non-convex optimization problem.

Specifically, we propose to learn the Bayesian Lasso linear model $\mathbf{y} = \mathbf{\Phi}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, with $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}|0, \sigma^2 I)$ (we assume σ^2 given in this work, and pick it's value from among a set of candidates based on predictive accuracy estimates such as cross validation/validation error). Further, $p(\boldsymbol{\beta}|\boldsymbol{\gamma}) = \mathcal{N}(\boldsymbol{\beta}|0, \Gamma)$ (recall that $\Gamma = \text{diag}[\boldsymbol{\gamma}]$) and we place an exponential prior on the variance components,

$$p(\boldsymbol{\gamma}|\boldsymbol{\lambda}) = \prod_{j=1}^{p} \frac{\lambda_j}{2} \exp -\frac{\lambda_j \gamma_j}{2}.$$

However, as in SBL, we choose to estimate the non-negative hyperparameters $\boldsymbol{\gamma}$ by type-II maximum likelihood. In other words, we maximize the marginal data likelihood in order to learn the hyperparameters:

$$\begin{aligned} p(\boldsymbol{\gamma}|\mathcal{D}, \boldsymbol{\lambda}) &\propto p(\mathbf{y}|\boldsymbol{\gamma}) p(\boldsymbol{\gamma}|\boldsymbol{\lambda}) \\ &= \left(\int p(\mathbf{y}|\boldsymbol{\beta}) p(\boldsymbol{\beta}|\gamma) d\boldsymbol{\beta} \right) \; p(\boldsymbol{\gamma}|\boldsymbol{\lambda}). \end{aligned}$$

Taking the negative logarithm, using the result from Equation 3, and removing quantities irrelevant to the optimization problem results in the following objective function to be minimized:

$$\mathcal{L}(\boldsymbol{\gamma}) = \log |\Sigma_y| + \mathbf{y}^{\mathrm{T}} \Sigma_y^{-1} \mathbf{y} + \lambda \sum_{j=1}^{p} \gamma_j \tag{5}$$

Note that for parsimony and convenience in further estimation, we set all the $\lambda_j = 2\lambda$, which we assume to be given (again picked from candidates using cross validation). Also note that the key difference compared to SBL is the presence of the proper variance prior, which results in the extra term in Equation 5 as compared to Equation 3, and provides extra shrinkage. After obtaining (local) maximum values for the hyperparameters $\boldsymbol{\gamma}_*$ (the next section outlines algorithms for this purpose), we then make posterior predictions also according to the SBL machinery, via the expressions for $p(y_t|\mathcal{D}, \boldsymbol{\gamma}_*, \sigma^2, \phi(\mathbf{x}_t))$ and the related expressions for the mean and variance, Equations 4. We call this approach the demi-Bayesian Lasso (dBL).

It is worth mentioning that the above formulation can be obtained by considering the original SBL formulation with a particular form of the Gamma prior on the variance components γ_j. This links the Bayesian Lasso model to the SBL model and provides the motivation for our proper prior on the variances.

3.1 Algorithms

The key learning task with the model is finding optimal prior variance, $\boldsymbol{\gamma}$ values. This then allows us to compute the posterior distribution over the weights and compute the posterior predictive distribution (Equations 4). Due to the similarity with the SBL objective function, many of the SBL algorithms apply with minor modifications. Here we discuss two variants. The first is a modification of the EM algorithm that was proposed in Tipping (2001). Starting with some $\boldsymbol{\gamma}$, we iteratively apply the E step:

$$\Sigma = \Gamma - \Gamma \boldsymbol{\Phi}^{\mathrm{T}} \Sigma_{y*}^{-1} \boldsymbol{\Phi} \Gamma,$$

with $\boldsymbol{\mu} = \Gamma_* \boldsymbol{\Phi}^{\mathrm{T}} \Sigma_{y*}^{-1} \mathbf{y}$ and the M step:

$$\gamma_j = \frac{2(\mu_j^2 + \Sigma_{jj})}{1 + \sqrt{1 + 4\lambda(\mu_j^2 + \Sigma_{jj})}},$$

for all $j = 1, \ldots, p$, until convergence. We will refer to this algorithm as EM dBL.

The second variant modifies a recent algorithm by Wipf and Nagarajan (2008) that possesses several nice properties, such as a global convergence analysis (to a local minimum) and sparsity along the solution path. We state the algorithm next, which we will call ℓ_1 dBL, followed by a brief deviation. We refer the reader to Wipf and Nagarajan (2008) for further details.

Data: $\mathcal{D}, \lambda, \gamma$.
Result: Sparse $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, at each iteration.
Initialize $\boldsymbol{\beta} = \mathbf{0}$, $\mathbf{z} = [1, \ldots, 1]^{\mathrm{T}}$.
while *Convergence criteria not met* **do**
  $\boldsymbol{\beta}_* = \mathrm{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2 + 2\sigma^2 \sum_j (z_j + \lambda)^{1/2} |\beta_j|$
  $\gamma_j = (z_j + \lambda)^{-1/2} |\beta_{*,j}|$
  $\mathbf{z}_* = \nabla_{\boldsymbol{\gamma}} \log |\Sigma_y| = \mathrm{diag}[\boldsymbol{\Phi}^{\mathrm{T}} \Sigma_y^{-1} \boldsymbol{\Phi}]$
  $\boldsymbol{\beta} = \boldsymbol{\beta}_*$
  $\mathbf{z} = \mathbf{z}_*$
end
$\boldsymbol{\beta} = E[\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\gamma}_*] = \Gamma_* \boldsymbol{\Phi}^{\mathrm{T}} \Sigma_{y*}^{-1} \mathbf{y}$

Algorithm 1: The ℓ_1 dBL algorithm.

The algorithm outlined above is guaranteed to converge monotonically to a local minimum or saddle point of Equation 5. This follows trivially

from Theorem 1 and analysis in Wipf and Nagarajan (2008). The algorithm notably uses iterated re-weighted ℓ_1 regression (step 1 in the while loop) to estimate the weights $\boldsymbol{\beta}$, also known as an adaptive Lasso problem (Zou, 2006). The ℓ_1 penalty results in sparse $\boldsymbol{\beta}$, which correspondingly results in sparse estimates of variance components $\boldsymbol{\gamma}$—we will refer to this algorithm as ℓ_1 dBL. The auxiliary variables $\mathbf{z}$ (a p-vector) arise from the upper bound of the log-determinant term (see 3.1.1). The choice of an Exponential prior results in very small computational difference between the SBL algorithm in Wipf and Nagarajan (2008) and the one presented here. In particular, replacing $z_j + \lambda$ with z_j is the only difference. Similarly, the prior results in a small difference in the M step in the corresponding update in Tipping (2001) algorithm, where it is: $\gamma_j = \mu_j^2 + \Sigma_{jj}$. As expected, the proper prior results in additional regularization of the variance parameters towards zero. We expect that this additional regularization will come with a bias-variance trade-off, the additional flexibility created by the single extra parameter λ potentially allowing us to generalize better.

3.1.1 Deriving the ℓ_1 dBL algorithm

Here we briefly outline the algorithm derivation. The log-determinant term in $\mathcal{L}(\boldsymbol{\gamma})$ (Eq. 5) is concave in $\boldsymbol{\gamma}$, and so can be expressed via

$$\log|\Sigma_y| = \min_{\mathbf{z}} \mathbf{z}^{\mathrm{T}}\boldsymbol{\gamma} - g^*(\mathbf{z}).$$

In that expression, $g^*(\mathbf{z})$ is the concave conjugate of $\log|\Sigma_y|$, $g^*(\mathbf{z}) = \min_{\boldsymbol{\gamma}} \mathbf{z}^{\mathrm{T}}\boldsymbol{\gamma} - \log|\Sigma_y|$. This then leads to the upper bounding cost function:

$$\mathcal{L}(\boldsymbol{\gamma}, \mathbf{z}) = \mathbf{z}^{\mathrm{T}}\boldsymbol{\gamma} - g^*(\mathbf{z}) + \mathbf{y}^{\mathrm{T}}\Sigma_y^{-1}\mathbf{y} + \lambda\sum_{j=1}^{p}\gamma_j \geq \mathcal{L}.$$

Following Wipf and Nagarajan (2008), the optimal $\mathbf{z}$ occurs when

$$\mathbf{z}_* = \nabla_{\boldsymbol{\gamma}}\log|\Sigma_y| = \mathrm{diag}[\boldsymbol{\Phi}^{\mathrm{T}}\Sigma_y^{-1}\boldsymbol{\Phi}].$$

Re-expressing the term

$$\mathbf{y}^{\mathrm{T}}\Sigma_y^{-1}\mathbf{y} = \min_{\boldsymbol{\beta}} \frac{1}{\sigma^2}\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2 + \sum_j \frac{\beta_j^2}{\gamma_j},$$

we get an upper bounding term

$$\mathcal{L}_{\mathbf{z}}(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \frac{1}{\sigma^2}\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^{p}\left((z_j + \lambda)\gamma_j + \frac{\beta_j^2}{\gamma_j}\right) \geq \mathcal{L}$$

which is jointly convex in $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, which can be globally minimized solving for $\boldsymbol{\gamma}$ and then $\boldsymbol{\beta}$ (Wipf and Nagarajan, 2008). Now, for any $\boldsymbol{\beta}$, $\gamma_j = (z_j + \lambda)^{-1/2}|\beta_j|$ minimizes $\mathcal{L}_{\mathbf{z}}(\boldsymbol{\gamma}, \boldsymbol{\beta})$. This then results in algorithm 1 which is an iterative application of the steps of finding the optimal $\boldsymbol{\gamma}$ (minimizing the upper bounding cost), and then finding the optimal $\mathbf{z}$ (which then leads to recomputing the optimal upper bounding cost).

3.2 An EN heuristic

While the use of of iterated re-weighted ℓ_1 regularized regression results in sparsity which is desirable, it also inherits some of the drawbacks of ℓ_1 regression. In particular, an issue of concern is the instability of ℓ_1 regression solutions with respect to highly correlated regressors (Zou and Hastie, 2005). Essentially, with highly correlated regressors/basis functions, the weights $\boldsymbol{\beta}$ computed based on the ℓ_1 solution are unstable—small differences in the dataset can result in the selection of very different subsets of a set of correlated regressors[3]. Zou and Hastie (2005)'s "elastic net" seeks to address this issue. The elastic net imposes an $\alpha\ell_1 + (1-\alpha)\ell_2$ penalty, $0 \leq \alpha < 1$, on the weights. This has the attractive property of imposing a simple additional convex loss and encourages a "grouping effect" which helps keep weights on correlated regressors similar (ref Thm. 1 in Zou and Hastie 2005). Zou and Hastie (2005) show good results when applying this mixed penalty.

We attempt to capture the same effect in the dBL. This is done by solving an elastic net problem in Algorithm 1 instead of the re-weighted ℓ_1 regression problem. Unfortunately, the heuristic doesn't correspond to an intuitive prior on the variance components and further is strongly tied to the iterated re-weighted ℓ_1 regression algorithm (an equivalent is hard to define for the EM style algorithms). Nonetheless, we explore this heuristic in the experiments that follow—we will refer to this as dBL+EN below.

[3]For two perfectly correlated relevant regressors, one amongst them is chosen to have a non-zero weight either at random or due to the particulars of the the algorithm implementation.

4 Experiments and Results

We now turn to evaluation of the dBL via experimental studies. We consider both simulation studies and three real data examples from the literature and evaluate the strengths and weaknesses of the proposal.

4.1 Simulation studies

Our simulation study models are based on the studies in Zou and Hastie (2005) (examples 2 and 4 correspond exactly, examples 1 and 3 are minor modifications of examples in their work). The aim is to highlight the differences between the techniques in terms of predictive performance, but also in terms of variable selection accuracy. We present five simulation study examples, each of which consist of a training set, a validation set and a test set (all independent). Models are fit using the training data only, and parameters/hyperparameters selected from appropriate grids on reasonable values using the validation set. For the EN heuristic, in all experiments we set the ℓ_1/ℓ_2 blending parameter $\alpha = 0.7$. Borrowing notation from Zou and Hastie (2005), we use x/y/z to denote x training observations (size of the training data), y validation and z independent test samples. The four examples attempt to gauge the performance of the methods in various scenarios:

- Example 1: we simulate 200 data sets consisting of 20/20/200 observations with 8 predictors. The data generating mechanism is a linear model with $\mathbf{y} = \mathbf{\Phi}\boldsymbol{\beta} + \kappa\boldsymbol{\epsilon}$ where $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}|0, I)$ and $\kappa = 3$. We set $\boldsymbol{\beta} = [3, 1.5, 0, 0, 2, 0, 0, 0]^{\mathrm{T}}$. The pairwise correlation between $\mathbf{\Phi}_i$ and $\mathbf{\Phi}_j$ is set as $cov(i, j) = \rho^{|i-j|}$. In example 1, the covariance matrix is an identity matrix, $cov(i, j) = 0$ for all $i \neq j$ and $cov(i, i) = 1$. Finally, $\mathbf{\Phi}$ is drawn from a multivariate Gaussian with zero mean and the above covariance matrix.
- Example 2: Is entirely analogous to example 1 except with non-identity covariance (introducing mild correlation between the regressors). Here, $\rho = 0.5$.
- Example 3: Is the same as examples 1 and 2, except with higher correlation between the regressors. Here, $\rho = 0.85$.
- Example 4: Also an example where the data generating mechanism is a linear model. We simulate 200 data sets with 100/100/400 observations and 40 predictors. This time $\boldsymbol{\beta} = [0, \ldots, 0, 2, \ldots, 2, 0, \ldots, 0, 2, \ldots, 2]^{\mathrm{T}}$, with alternating blocks of 10 indices of zeros and 2s. Here, $\kappa = 15$ and $cov(i, j) = 0.5$ for all $i \neq j$, and $cov(i, i) = 1$.

• Example 5: An example where the data generating mechanism is *not* a linear model. Here we will not be able to gauge variable selection accuracy, but only predictive performance. In this case we include some interaction terms and powers of the regressors in computing the response. We simulate 40/40/400 observations following polynomial model (for a single observation): $y = 1.5\phi_1^2 + 2\phi_1\phi_2 - \phi_5\phi_1 + \phi_5^3 + 2\phi_7 + 3\epsilon$ where ϵ is a zero mean, unit variance Gaussian error. The learning algorithms only get access to $\mathbf{\Phi}$ and the responses.

For examples 1 through 4, we compute the following quantities: i) the mean squared error (MSE), computed on test data, ii) mean "parametric" error (MPE), that is, the mean of the quantity $(\boldsymbol{\beta} - \boldsymbol{\beta}_{true})^{\mathrm{T}}\Sigma(\boldsymbol{\beta} - \boldsymbol{\beta}_{true})$, where Σ is the covariance of $\mathbf{\Phi}$. This attempts to quantify closeness to the parameters that actually generated the data. iii) Quantities related to structural errors: mean C ($\bar{C}$) and mean IC ($\bar{IC}$). C is defined as the number of true weights that were zero which are correctly estimated as zero by the model (thus higher values are better). Similarly, IC is defined as the number of non-zero true weights incorrectly estimated as zero by the model (and thus lower IC values are preferred). Models that are excessively sparse would tend to have high C values (good) and high IC values (not good). A model that is completely non-sparse would have the lowest possible C value (bad) but the lowest IC values (good) as well. For example 5, since the data generating mechanism is outside the model hypothesis class we only report the test mean squared error.

We evaluate the optimization based approaches, namely the Lasso (Lasso in the results), the original SBL algorithm (Tipping 2001, SBL), the Wipf and Nagarajan (2008) SBL algorithm (ℓ_1 SBL), the dBL model with parameters found using the EM algorithm (EM dBL) and the ℓ_1 variation (Algorithm 1, ℓ_1 dBL) and finally, the ℓ_1 based proposal with the EN heuristic (dBL + EN).

Table 1 and Figure 2 show the results. In all cases (modest to large) improvements are made over the flat-prior variants and over the Lasso both in terms of prediction accuracy as well as structural accuracy. In the tables we show standard errors of the estimates, and in the Figure, we show boxplots of the squared error showing the median, lower and upper quartiles, whiskers and outliers. We next turn to some real data examples.

Table 1: Simulation study results

	Lasso	**SBL**	ℓ_1 **SBL**	**EM dBL**	ℓ_1 **dBL**	**dBL + EN**
Example 1						
MSE	14.40 (0.28)	14.39 (0.31)	14.66 (0.34)	14.23 (0.29)	**14.04** (0.30)	14.11 (0.28)
MPE	3.99 (0.21)	4.02 (0.24)	4.25 (0.27)	3.83 (0.22)	**3.64** (0.23)	3.70 (0.21)
$\bar{C}$	2.23 (0.12)	2.59 (0.12)	3.51 (0.10)	2.21 (0.11)	**3.61** (0.10)	3.29 (0.09)
$I\bar{C}$	0.24 (0.04)	0.26 (0.04)	0.38 (0.05)	**0.22** (0.04)	0.30 (0.04)	0.26 (0.04)
Example 2						
MSE	14.63 (0.36)	14.84 (0.37)	15.12 (0.42)	14.44 (0.36)	14.42 (0.36)	**14.22** (0.37)
MPE	3.91 (0.22)	4.12 (0.23)	4.44 (0.30)	3.72 (0.21)	3.72 (0.21)	**3.53** (0.22)
$\bar{C}$	2.24 (0.11)	2.77 (0.11)	3.56 (0.11)	2.23 (0.10)	3.58 (0.10)	**3.25** (0.10)
$I\bar{C}$	**0.22** (0.03)	0.39 (0.04)	0.48 (0.05)	**0.22** (0.03)	0.36 (0.04)	0.23 (0.03)
Example 3						
MSE	14.20 (0.32)	14.42 (0.31)	15.20 (0.42)	13.83 (0.30)	13.99 (0.30)	**13.44** (0.28)
MPE	3.33 (0.17)	3.56 (0.17)	4.21 (0.29)	2.96 (0.15)	3.09 (0.15)	**2.53** (0.13)
$\bar{C}$	2.42 (0.09)	2.85 (0.10)	3.23 (0.09)	2.52 (0.09)	**3.48** (0.09)	2.77 (0.08)
$I\bar{C}$	0.65 (0.05)	0.78 (0.05)	0.91 (0.05)	0.58 (0.05)	0.92 (0.05)	**0.38** (0.04)
Example 4						
MSE	316.92 (2.41)	311.37 (2.32)	327.13 (2.78)	283.72 (1.95)	286.50 (2.03)	**261.14** (1.67)
MPE	83.74 (1.64)	77.37 (1.39)	93.55 (2.02)	49.28 (0.90)	52.19 (1.03)	**26.22** (0.49)
$\bar{C}$	9.72 (0.34)	14.61 (0.24)	8.39 (0.16)	11.99 (0.19)	**14.66** (0.18)	8.03 (0.21)
$I\bar{C}$	5.92 (0.19)	9.87 (0.19)	5.79 (0.13)	6.58 (0.14)	8.77 (0.15)	**2.52** (0.12)
Example 5						
MSE	30.78 (0.40)	30.56 (0.42)	31.64 (0.47)	30.32 (0.40)	**30.07** (0.40)	30.37 (0.40)

4.2 Prostate cancer data

The data in this example comes from a prostate cancer study done by Stamey et al. (1989). Eight clinical measurements serve as the regressors, which are, in order: log(cancer volume) *lcavol*, log(prostate weight) *lweight*, *age*, log(amount of benign prostatic hyperplasia) *lbph*, seminal vesicle invasion *svi*, log(capsular penetration) *lcp*, Gleason score *gleason* and percentage Gleason score 4 or 5 *pgg*45. The predictive quantity of interest is the log(prostate specific antigen) *lpsa*.

Following Zou and Hastie (2005), we divide the data into two parts, a training set with roughly two thirds the number of observations, 64 observations and a test set with 33 observations. Hyperparameters were selected from a grid of values via 10-fold cross validation using only the training data[4]. The methods are compared via the prediction mean-squared error on the test

[4]For all the real data examples, we select the hyperparameters following the slight modification to k-fold CV suggested in Chapter 7 of Hastie et al. (2001), namely we pick the largest amount of regularization that is within 1 standard error of the minimum CV error.

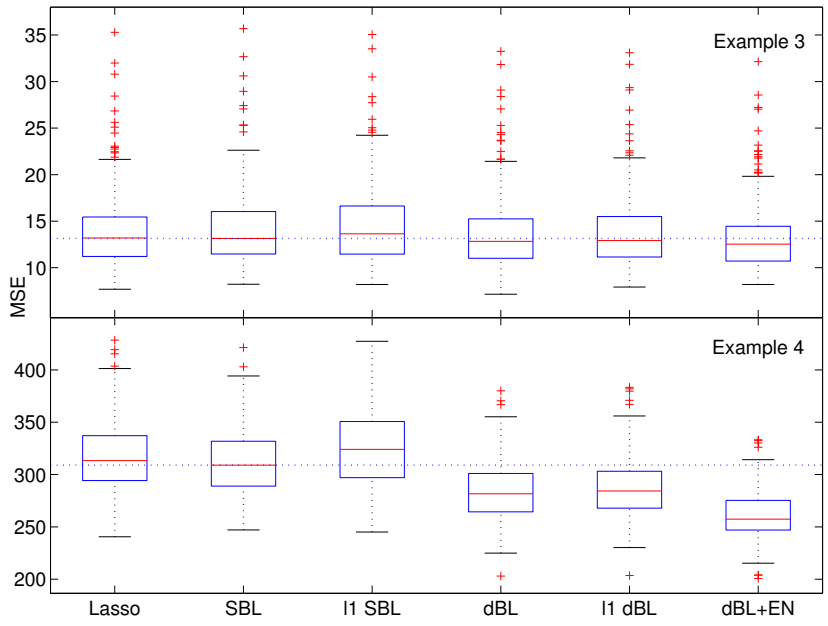

Figure 2: Boxplots for simulation studies 3 and 4. The horizontal dashed line is a visual guide and marks the location of the minimum median from amongst the prior art, namely the Lasso, SBL and ℓ_1 SBL.

data.

Our results (Table 2) show improved performance of the proposals over the Lasso[5] and SBL, with the ℓ_1 dBL providing the best performance. There is broad consensus on the selected variables, with *lcp* being rejected by all models in our experiments.

4.3 Diabetes data

The data in this study come from Efron et al. (2004). The response is a quantitative measure of diabetes progression in 398 patients one year after baseline. The predictors include age, sex, body mass index, average blood pressure, and six blood serum measurements, for a total of 10 regressors.

[5]Note that in the table, the σ^2 is a proxy label for the regularization parameter for the Lasso.

Table 2: Prostate data results

	Lasso	**SBL**	ℓ_1 **SBL**	**EM dBL**	ℓ_1 **dBL**	**dBL + EN**
MSE	0.4505	0.5539	0.5765	0.5367	**0.3781**	0.5216
Vars	all	all but 6	all but 6	all but 6	(1,3,4,5)	(1,5,7)
σ^2 (λ)	0.5	0.2	0.01	0.25 (1)	0.005 (500)	0.25 (1)

As Efron et al. (2004) point out, linear models are especially useful in this diagnostic application, because in addition to predictive accuracy for future patients, the models would ideally provide disease progression guidance by being interpretable. We standardized the regressors to have zero mean 0 and unit variance.

We partition the data into a 266 patient training sample and a 132 patient test sample. Hyperparameters were selected from a grid of values via 10-fold cross validation using only the training data. We show test mean squared error, variables selected and parameters (and hyperparameters used).

Table 3: Diabetes data results

	Lasso	**SBL**	ℓ_1 **SBL**	**EM dBL**	ℓ_1 **dBL**	**dBL + EN**
MSE	3031.2	3045.1	3032.4	3034.2	3031.2	**3029.3**
Vars	all but 1,6,8	all but 1,7	all	all but 1,8	all but 1,6,8	all
σ^2 (λ)	1	500	500	500 (0.001)	100 (0.1)	500 (0.001)

Our results agree with many reported findings on this dataset, and in our experiments, the dBL + EN variant proved predictively best by a slight margin (Table 3). In terms of variable selection, the least important regressors appear to be 1, 6 and 8, which is also evident from the findings in Park and Casella (2008) (Note that in our experiments, the SBL model seems to deselect regressor 7, which is an anomaly).

4.4 Biscuit NIR data

In this application, we examine the biscuit dough data from (Brown et al., 1999). The response we look at is fat content of the dough (centered), and the regressors are spectral characteristics of the dough, measured using near infrared (NIR) spectroscopy (standardized). The spectral characteristics are described using a grid of wavelengths, in particular reflectance measured at every 4nm from the range of wavelengths: 1202—2400 nm. The data is split into 39 training samples and 31 test samples, and we standardize the regressors.

Hyperparameters were selected from a grid of values via 5-fold cross validation using only the training data. The methods are compared via the prediction mean-squared error on the test data.

Table 4: Biscuit NIR data results

	Lasso	**SBL**	**ℓ_1 SBL**	**EM dBL**	**ℓ_1 dBL**	**dBL + EN**
MSE	0.0565	0.0551	0.0696	0.0543	**0.0450**	0.1001
|Non-zero Vars|	18	6	269	11	54	43
σ^2 (λ)	1.25	0.25	0.05	0.2 (0.1)	0.15 (0.2)	1 (1)

Our results (Table 4, Figure 3) are consistent with previous studies that use this data (West, 2003) and we find ℓ_1 dBL gives the best performance. In particular, the non-zero $\boldsymbol{\beta}$ found by ℓ_1 dBL around 1710 nm are significant because fat is known to have a characteristic absorbance in this range. Also note that for this example, the dBL + EN heuristic appears to perform worse than the others.

5 Discussion

In this paper we examined the use of proper priors in sparse Bayesian learning and showed some promising experimental results. We show that with a single additional hyperparameter (set through cross-validation), the model is augmented substantially enough to make better predictions. Further, the choice of an exponential distribution as a prior connects SBL to the recently proposed Bayesian Lasso, with our proposal amounting to an attractive al-

ternative way of estimating Bayesian Lasso model hyperparameters by maximizing marginal likelihood rather than Monte Carlo simulation. We also explored the use of an EN-heuristic that, in our experiments, leads to better performance in the presence of correlated regressors. In future work we would like to extend the proposals to classification problems. We would also like to examine the efficient SBL algorithm of Tipping and Faul (2003) to see if an analogous procedure can be applied in this case as well. Finally, other forms of prior distribution on the variance are the topic of our further exploration - including additionally sparsifying priors like the Laplace distribution etc.

References

J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics, second edition edition, 1980.

P. J. Brown, T. Fearn, and M. Vannucci. The choice of variables in multivariate regression: a non-conjugate Bayesian decision theory approach. *Biometrika*, 86:635 – 648, 1999.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407 – 499, 2004.

A. Genkin, D. D. Lewis, and D. Madigan. Large-scale bayesian logisitic regression for text categorization. *Technometrics*, 49:291–304, 2007.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning; Data Mining, Inference and Prediction.* Springer, New York, 2001.

D. J. C. Mackay. Bayesian interpolation. *Neural Computation*, 4:415 – 447, 1992.

T. Park and G. Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681 – 686, 2008.

T. Stamey, J. Kabalin, J. McNeal, I. Johnstone, F. Freiha, E. Redwine, and N. Yang. Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate ii: radical prostatectomy treated patients. *Journal of Urology*, 16:1076 – 1083, 1989.

R. J. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

M. E. Tipping. Sparse Bayesian learning and the Relevance Vector Machine., 2001.

M. E. Tipping and A. C. Faul. Fast marginal likelihood maximisation for sparse Bayesian models. In C. M. Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, FL, 2003.

M. West. Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian Statistics*, 7(2003):723 – 732, 2003.

D. P. Wipf and S. Nagarajan. A New View of Automatic Relevance Determination. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, 2008.

H. Zou. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418 – 1429, 2006.

H. Zou and T. Hastie. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Society of Statistics (B)*, 67(2):301 – 320, 2005.

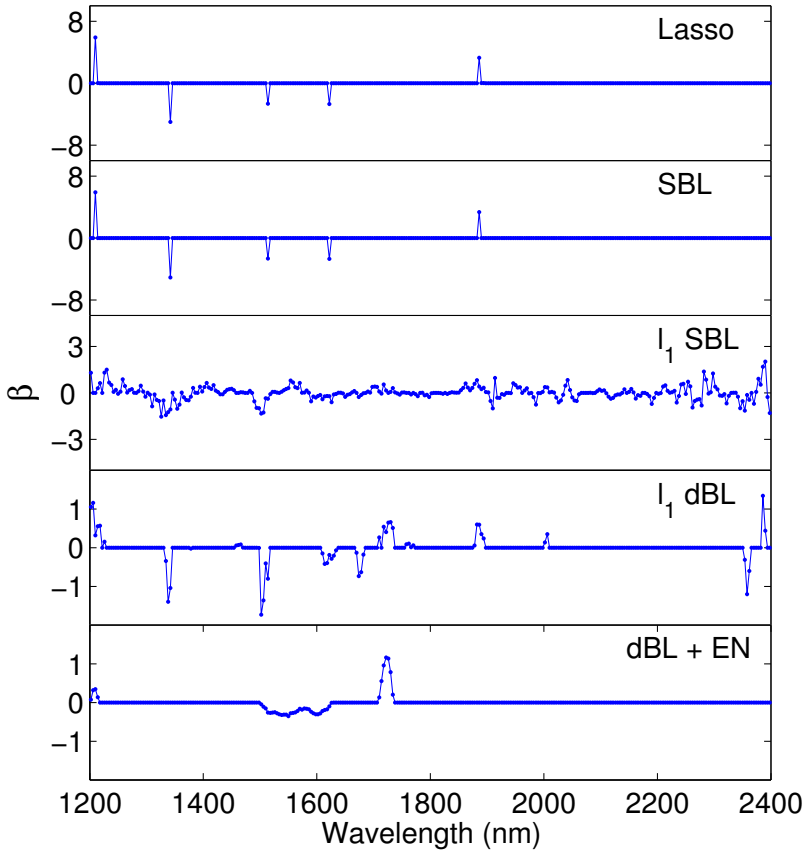

Figure 3: Biscuit data $\boldsymbol{\beta}$ values. Shown from top to bottom are the final parameter weights for the Lasso, SBL, ℓ_1 SBL, ℓ_1 dBL and dBL + EN. Due to the coarse resolution of the plot, only high magnitude weights can be discerned. Note the similarity between the high magnitude weights of the Lasso and SBL solutions (the EM SBL high magnitude weights are very similar and hence omitted).